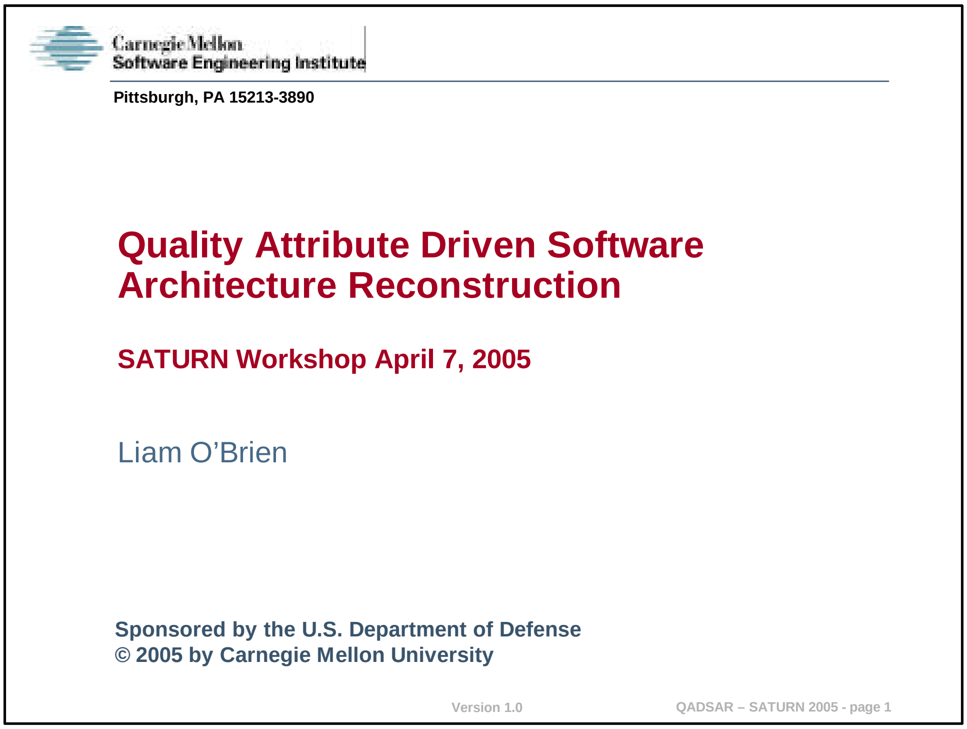


Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Quality Attribute Driven Software Architecture Reconstruction

SATURN Workshop April 7, 2005

Liam O'Brien

Sponsored by the U.S. Department of Defense

Motivation - 1

- Software architectures are critical to implement an organization's business goals and critical for intellectual property.

- Industry demands to:
  - Evolve existing products into Product Lines
  - Evaluate existing systems to improve response to quality attributes
  - Check conformance of the implementation to design
  - System Modernization

- Disconnect: Few organizations are willing to pay for an architecture reconstruction effort.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **07 APR 2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **Quality Attribute Driven Software Architecture Reconstruction**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
presentation from the SEI Software Architecture Technology User Network (SATURN) Workshop, April 6-7, 2005, Pittsburgh, PA.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**11**	

Motivation - 2

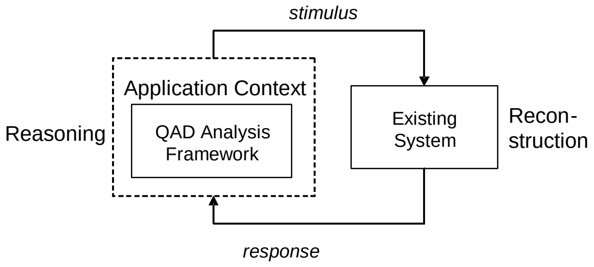
- Architecture Reconstruction is an embedded activity in a larger effort in an organization.
- Business goals are primarily incorporated as quality goals that shape the software architecture of a product.
- The analysis of software architectures is quality attribute driven.

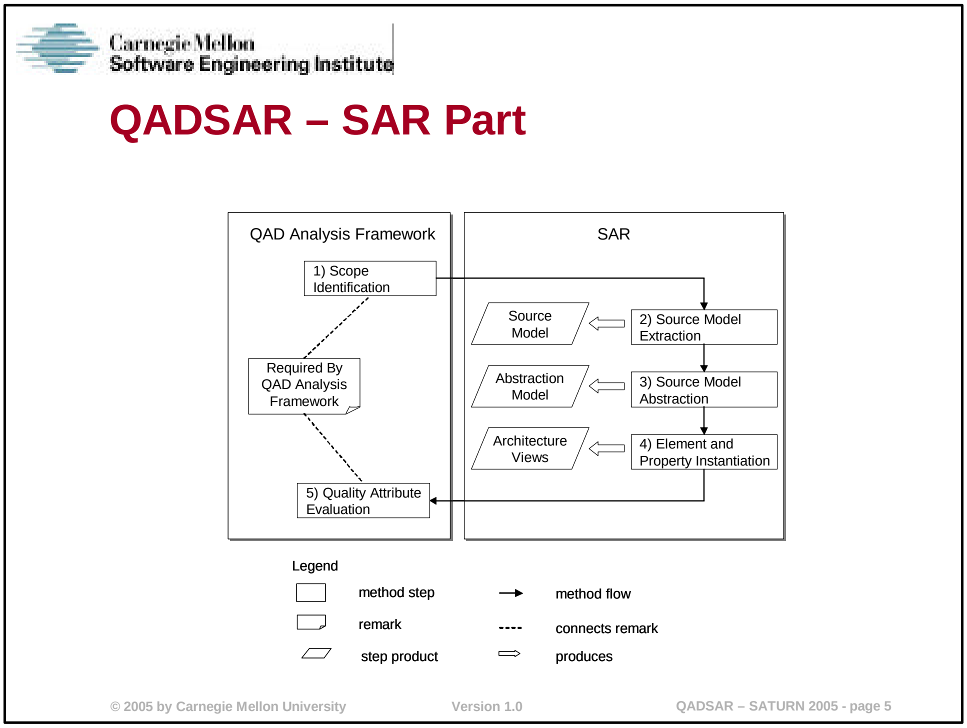
=> QADSAR
Quality Attribute Driven Software Architecture Reconstruction

QADSAR Approach

Characteristics

1. Goal-Driven Reconstruction (what to reconstruct?)
2. Stimulus/Response Approach
3. Requires Quality Attribute Reasoning Frameworks





QADSAR – SAR Part

QAD Analysis Framework

1) Scope Identification

Required By QAD Analysis Framework

5) Quality Attribute Evaluation

SAR

Source Model

2) Source Model Extraction

Abstraction Model

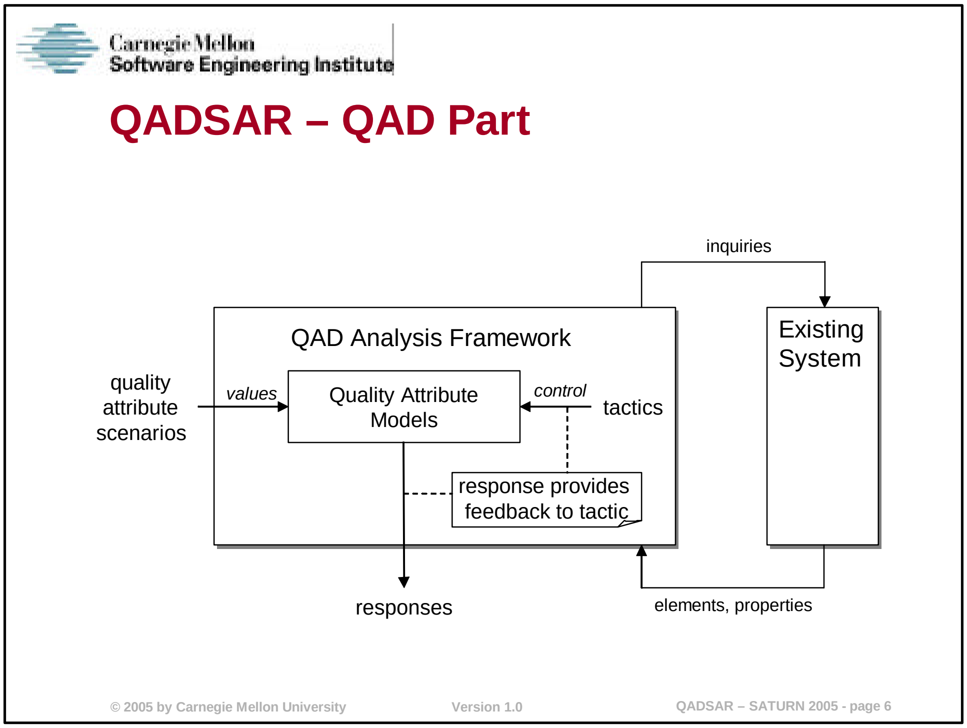
3) Source Model Abstraction

Architecture Views

4) Element and Property Instantiation

Legend

□	method step	→	method flow
□	remark	----	connects remark
▱	step product	⇨	produces



QADSAR – QAD Part

inquiries

QAD Analysis Framework

Existing System

quality attribute scenarios

values

Quality Attribute Models

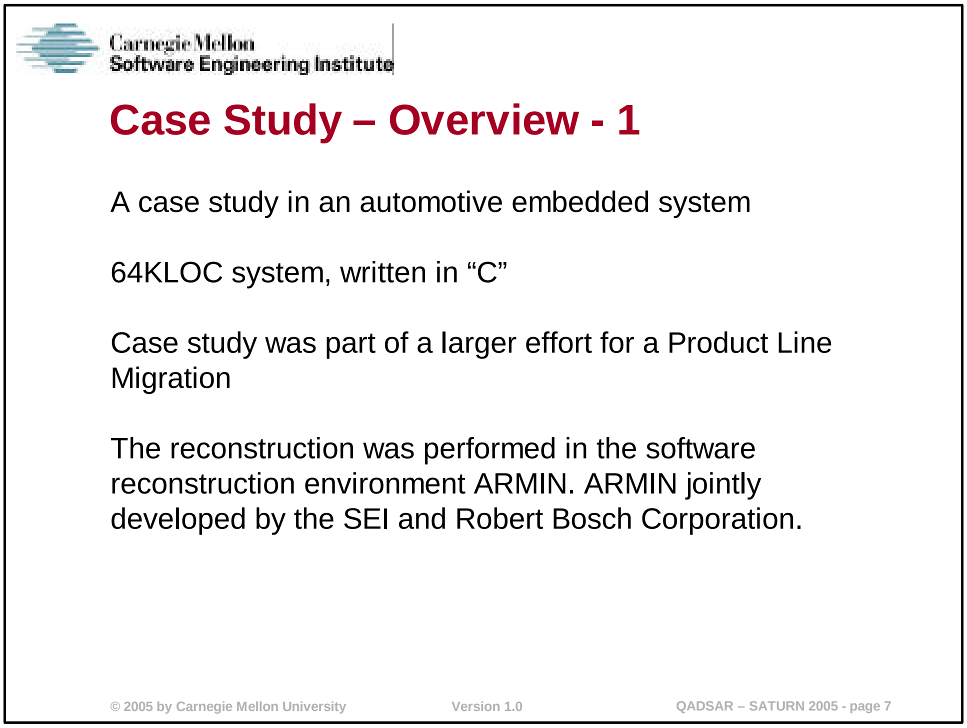
control

tactics

response provides feedback to tactic

responses

elements, properties



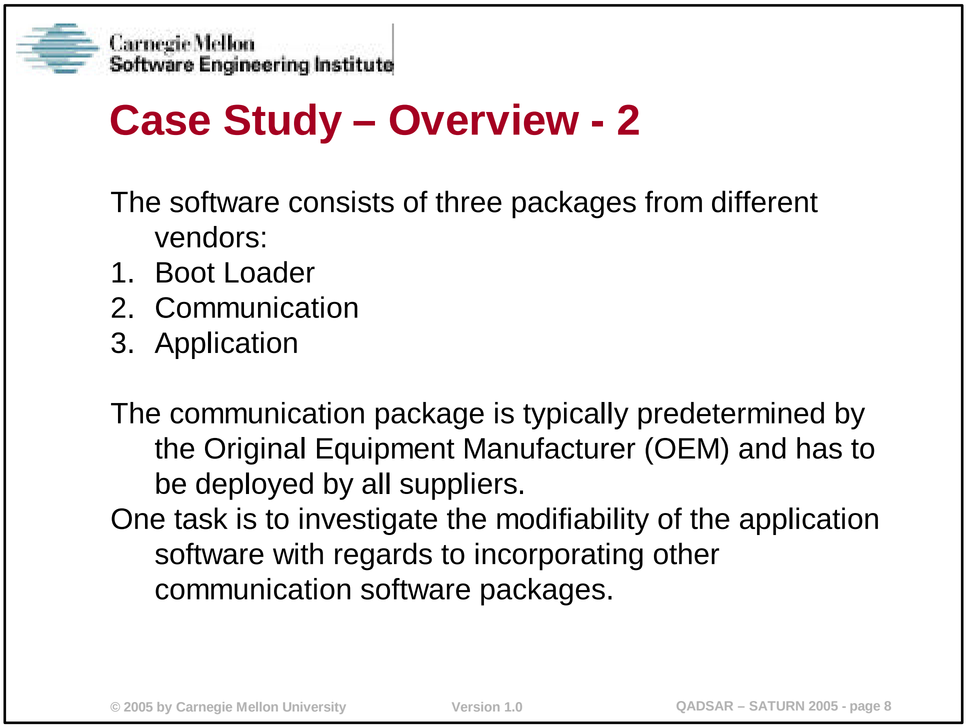
Case Study – Overview - 1

A case study in an automotive embedded system

64KLOC system, written in “C”

Case study was part of a larger effort for a Product Line Migration

The reconstruction was performed in the software reconstruction environment ARMIN. ARMIN jointly developed by the SEI and Robert Bosch Corporation.



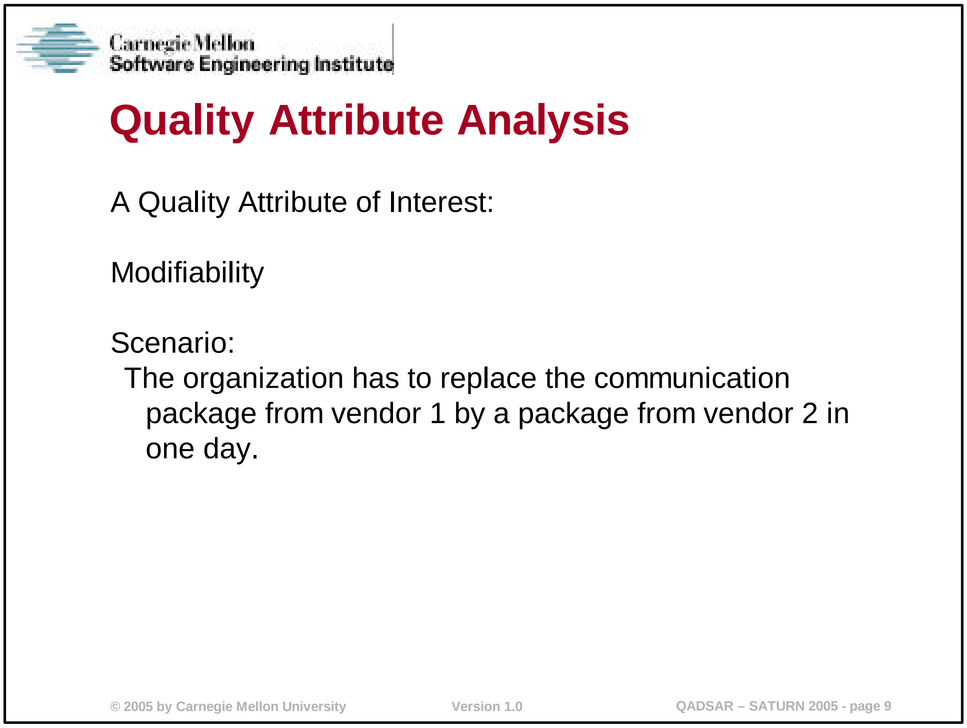
Case Study – Overview - 2

The software consists of three packages from different vendors:

1. Boot Loader
2. Communication
3. Application

The communication package is typically predetermined by the Original Equipment Manufacturer (OEM) and has to be deployed by all suppliers.

One task is to investigate the modifiability of the application software with regards to incorporating other communication software packages.



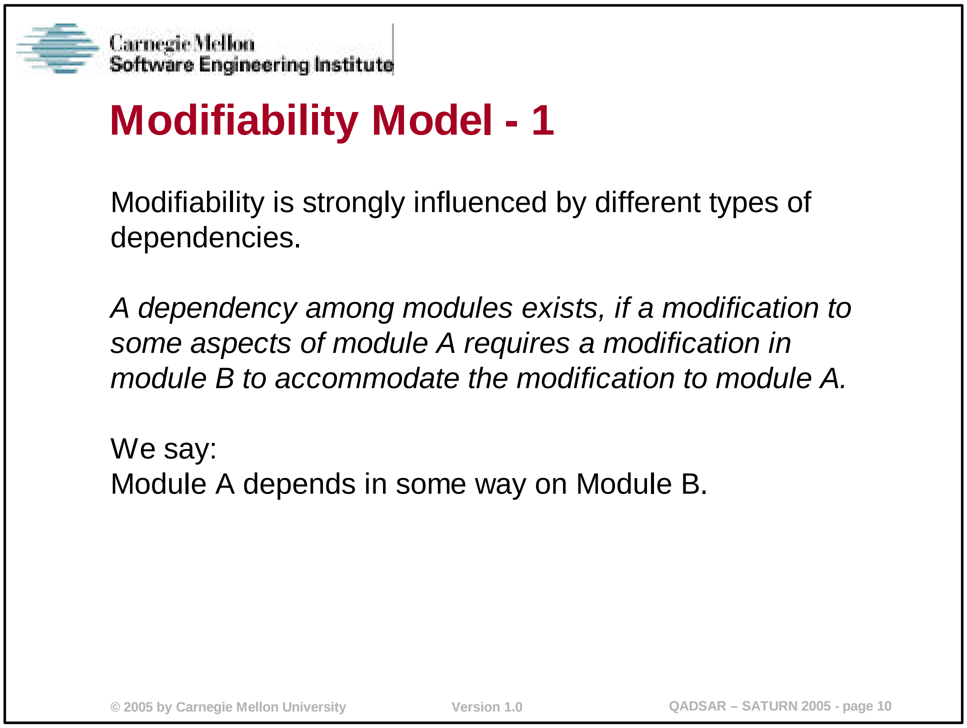
Quality Attribute Analysis

A Quality Attribute of Interest:

Modifiability

Scenario:

The organization has to replace the communication package from vendor 1 by a package from vendor 2 in one day.

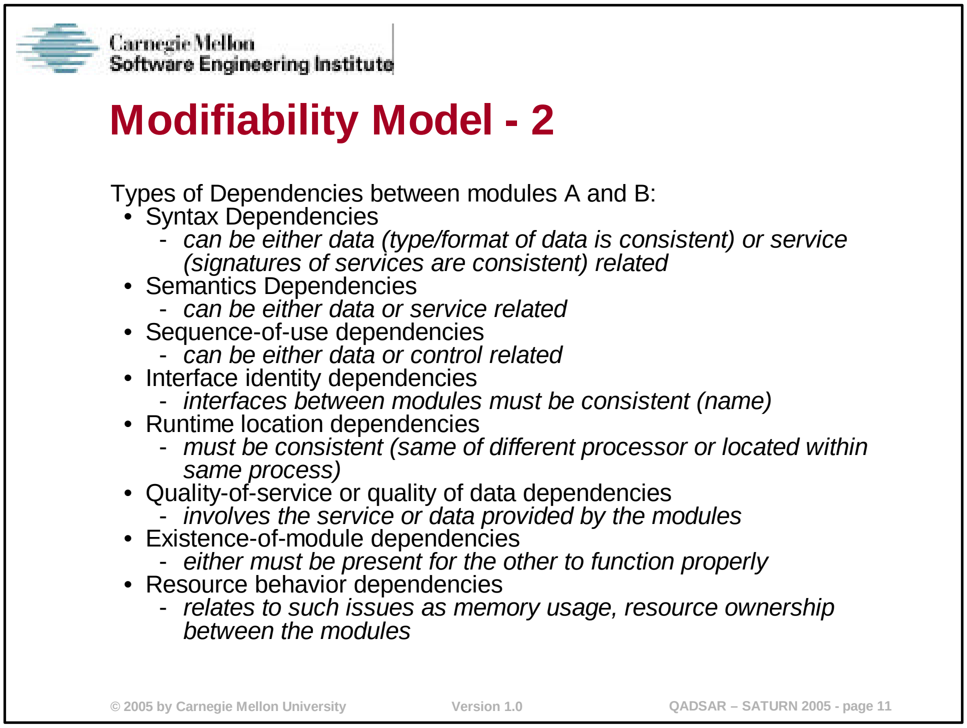


Modifiability Model - 1

Modifiability is strongly influenced by different types of dependencies.

A dependency among modules exists, if a modification to some aspects of module A requires a modification in module B to accommodate the modification to module A.

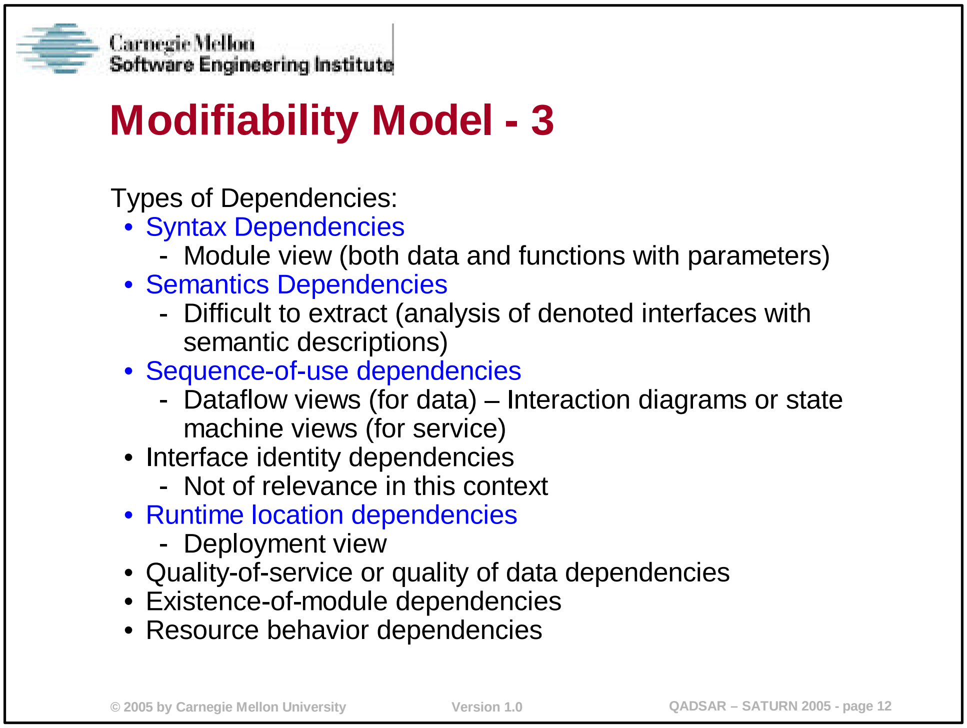
We say:
Module A depends in some way on Module B.



Modifiability Model - 2

Types of Dependencies between modules A and B:

- Syntax Dependencies
  - *can be either data (type/format of data is consistent) or service (signatures of services are consistent) related*
- Semantics Dependencies
  - *can be either data or service related*
- Sequence-of-use dependencies
  - *can be either data or control related*
- Interface identity dependencies
  - *interfaces between modules must be consistent (name)*
- Runtime location dependencies
  - *must be consistent (same of different processor or located within same process)*
- Quality-of-service or quality of data dependencies
  - *involves the service or data provided by the modules*
- Existence-of-module dependencies
  - *either must be present for the other to function properly*
- Resource behavior dependencies
  - *relates to such issues as memory usage, resource ownership between the modules*



Modifiability Model - 3

Types of Dependencies:

- Syntax Dependencies
  - Module view (both data and functions with parameters)
- Semantics Dependencies
  - Difficult to extract (analysis of denoted interfaces with semantic descriptions)
- Sequence-of-use dependencies
  - Dataflow views (for data) – Interaction diagrams or state machine views (for service)
- Interface identity dependencies
  - Not of relevance in this context
- Runtime location dependencies
  - Deployment view
- Quality-of-service or quality of data dependencies
- Existence-of-module dependencies
- Resource behavior dependencies

Modifiability Tactics

Modifiability might be achieved in a software system by different architecture tactics. Tactics for modifiability include:

- Maintain semantic coherence
- Isolate expected changes
- Hide information

Other Modifiability tactics include:

- Use a virtual machine
- Limit communications paths
- Abstract common services

Common to the tactics are strategies to reduce and manage the dependencies between modules.

Reconstruction for Modifiability

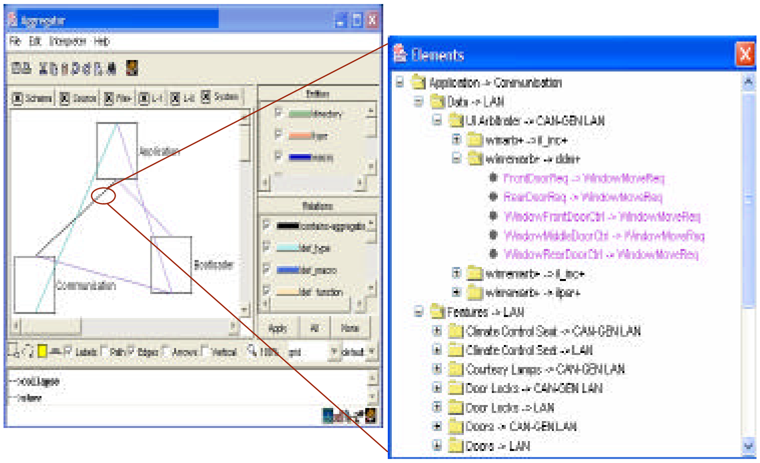
The tactics help us in the reconstruction process to identify what to look for in the system and to create particular views

Applied in this context:

- Maintain semantic coherence – did not evaluate because the communication package is already determined to be separated by the Original Equipment Manufacturer
- Isolate expected changes – is there something actively done in the application package to mitigate changes
- Hide information – e.g. is there an explicit model to separate interfaces from their realization

High-level Views of the Architecture

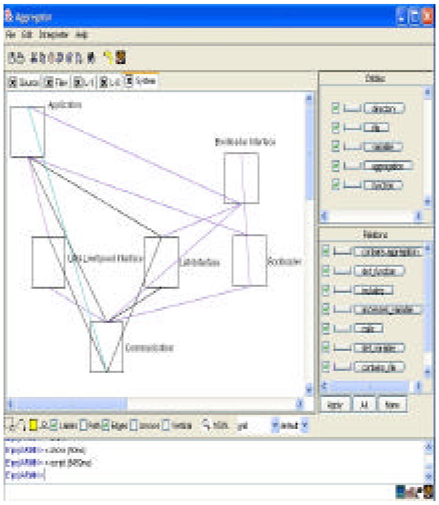
Reconstructed views of the architecture using ARMIN



- Application -> Communication
  - Data -> LAN
    - UI Arbitrater -> CAN-GEN LAN
      - wmarb+ -> il_inc+
      - wmremarb+ -> ddm+
        - FrontDoorReq -> WindowMoveReq
        - RearDoorReq -> WindowMoveReq
        - WindowFrontDoorCtrl -> WindowMoveReq
        - WindowMiddleDoorCtrl -> WindowMoveReq
        - WindowRearDoorCtrl -> WindowMoveReq
      - wmremarb+ -> il_inc+
      - wmremarb+ -> ipar+
  - Features -> LAN
    - Climate Control Seat -> CAN-GEN LAN
    - Climate Control Seat -> LAN
    - Courtesy Lamps -> CAN-GEN LAN
    - Door Locks -> CAN-GEN LAN
    - Door Locks -> LAN
    - Doors -> CAN-GEN LAN
    - Doors -> LAN

Module Views

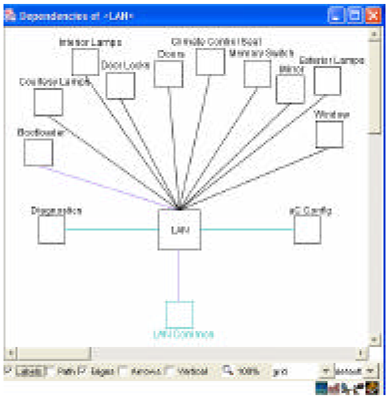
To detect the 'hide information' tactic, we isolate from the Application specific interface files that manage the access to and from the communication package



Module View – showing LAN

Response measure for the scenario – ‘number of modules affected’.

Generate a dependency layout for the LAN part of the communications package and analyze the application modules that access the LAN module.



Case Study Analysis

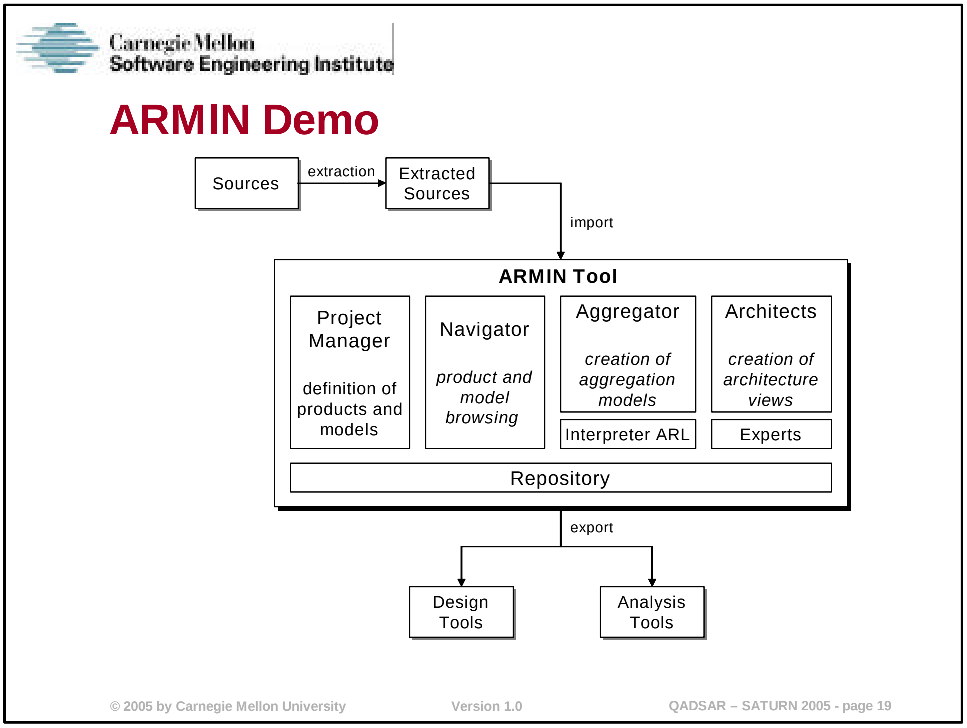
The two tactics ‘isolate expected changes’ and ‘hide information’ reduce the dependencies and the number of modules affected by changes.

Both tactics were not identifiable from our analysis

No explicit notion of interface identifiable – no separation of interface and implementation. Several attempts to capture an interface by different aggregations failed.

Analysis with the developers identified that changes are likely to be expected in more than a dozen modules.

Resulted in an improvement effort of the existing system to support the quality attribute scenario sufficiently.



ARMIN Demo

Sources

extraction

Extracted Sources

import

ARMIN Tool

Project Manager

definition of products and models

Navigator

product and model browsing

Aggregator

creation of aggregation models

Interpreter ARL

Architects

creation of architecture views

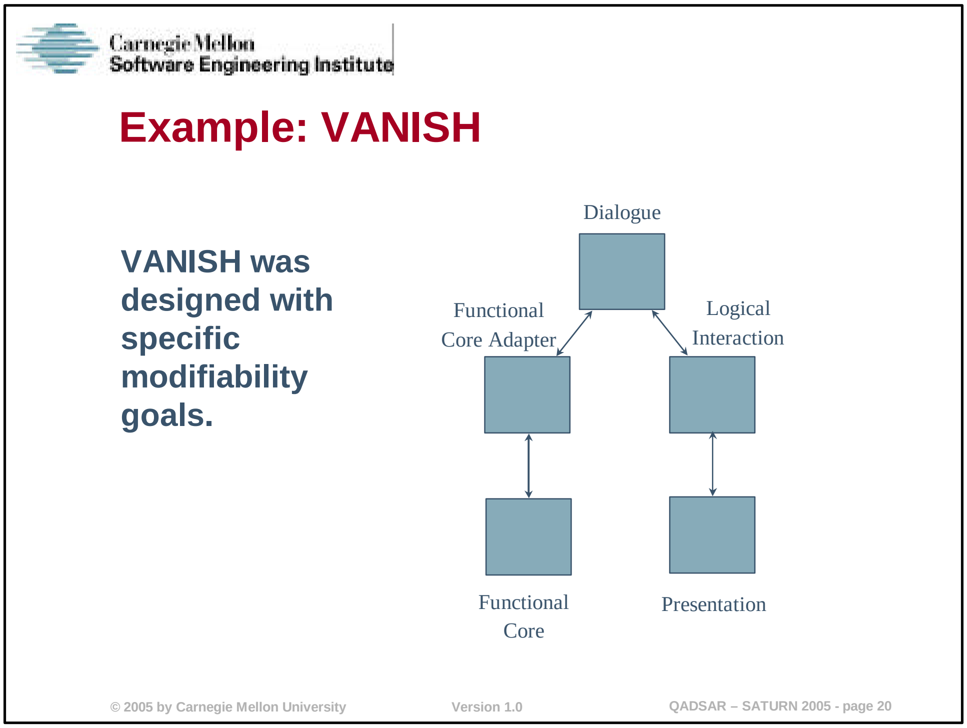
Experts

Repository

export

Design Tools

Analysis Tools



Example: VANISH

VANISH was designed with specific modifiability goals.

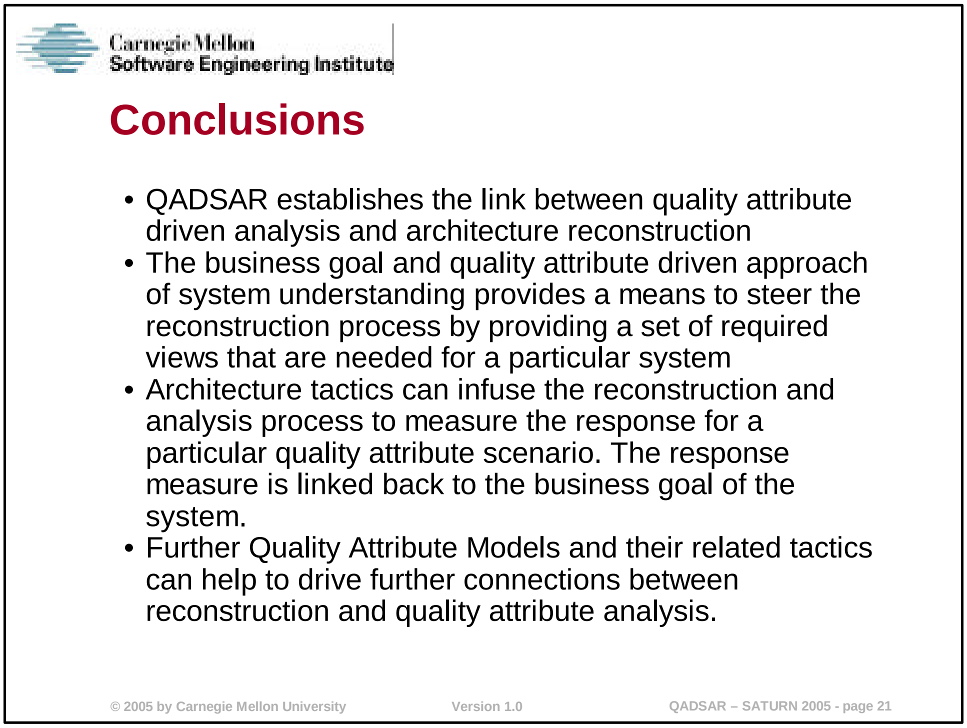
Dialogue

Functional Core Adapter

Logical Interaction

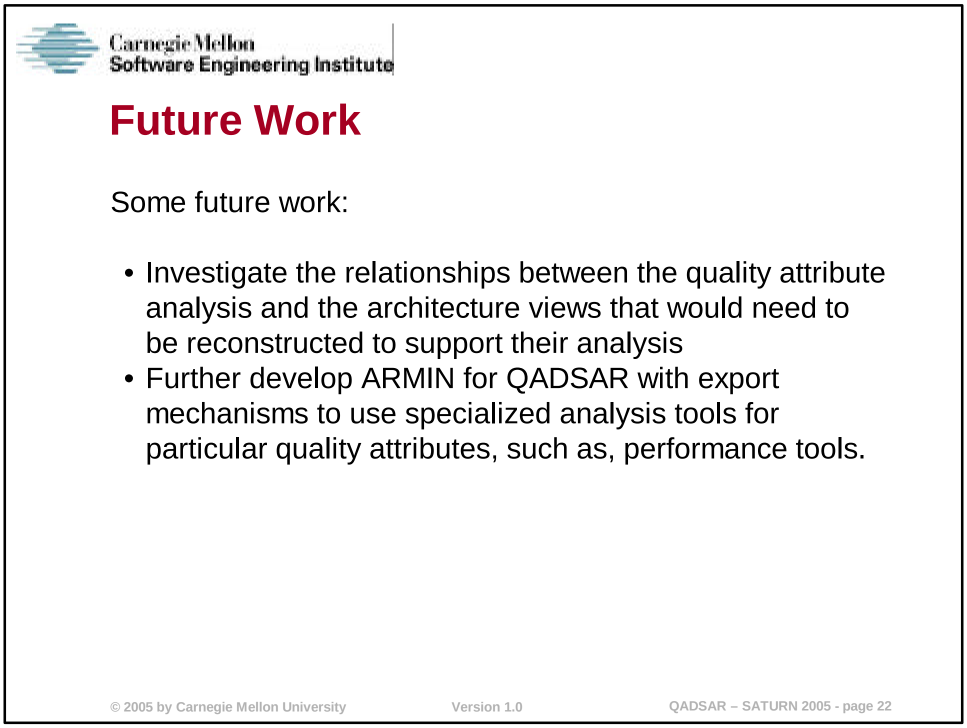
Functional Core

Presentation



Conclusions

- QADSAR establishes the link between quality attribute driven analysis and architecture reconstruction
- The business goal and quality attribute driven approach of system understanding provides a means to steer the reconstruction process by providing a set of required views that are needed for a particular system
- Architecture tactics can infuse the reconstruction and analysis process to measure the response for a particular quality attribute scenario. The response measure is linked back to the business goal of the system.
- Further Quality Attribute Models and their related tactics can help to drive further connections between reconstruction and quality attribute analysis.



Future Work

Some future work:

- Investigate the relationships between the quality attribute analysis and the architecture views that would need to be reconstructed to support their analysis
- Further develop ARMIN for QADSAR with export mechanisms to use specialized analysis tools for particular quality attributes, such as, performance tools.